

FIG. 1

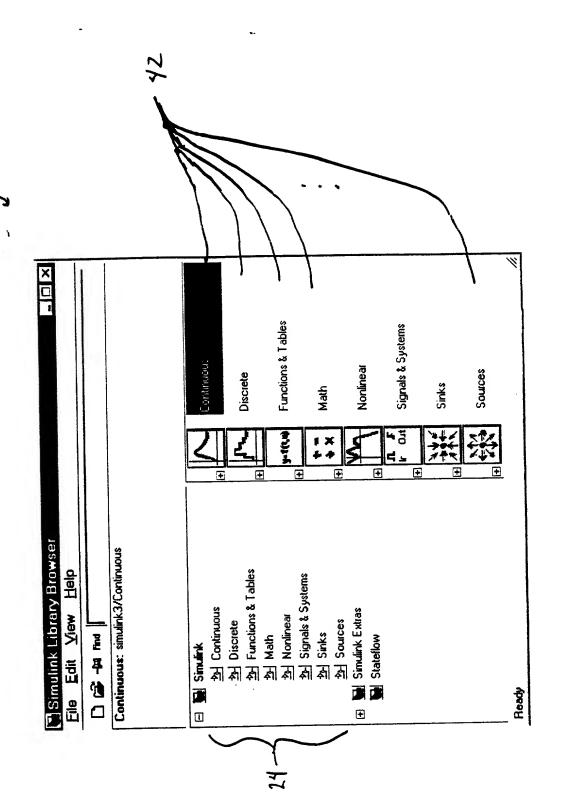


FIG. 2A

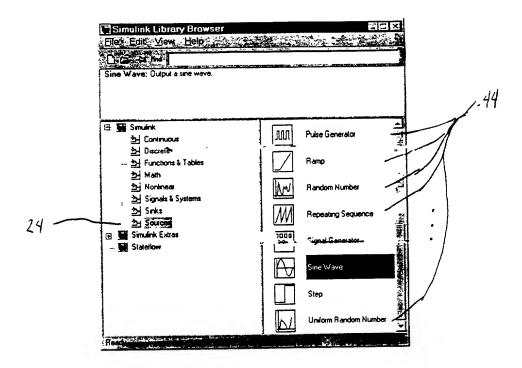


FIG. 2B

50

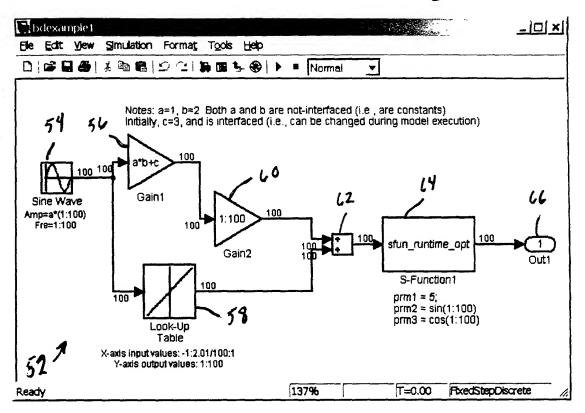


FIG. 3

s 70

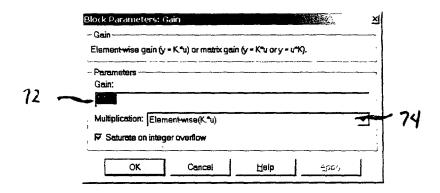


FIG. 4

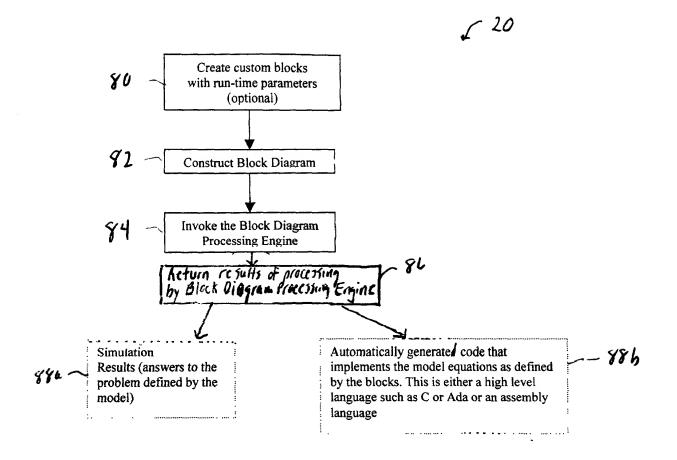


FIG. 5

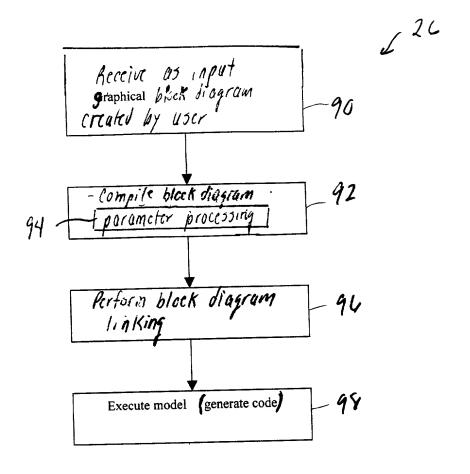


FIG. 6

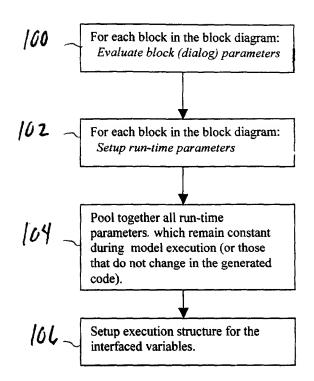
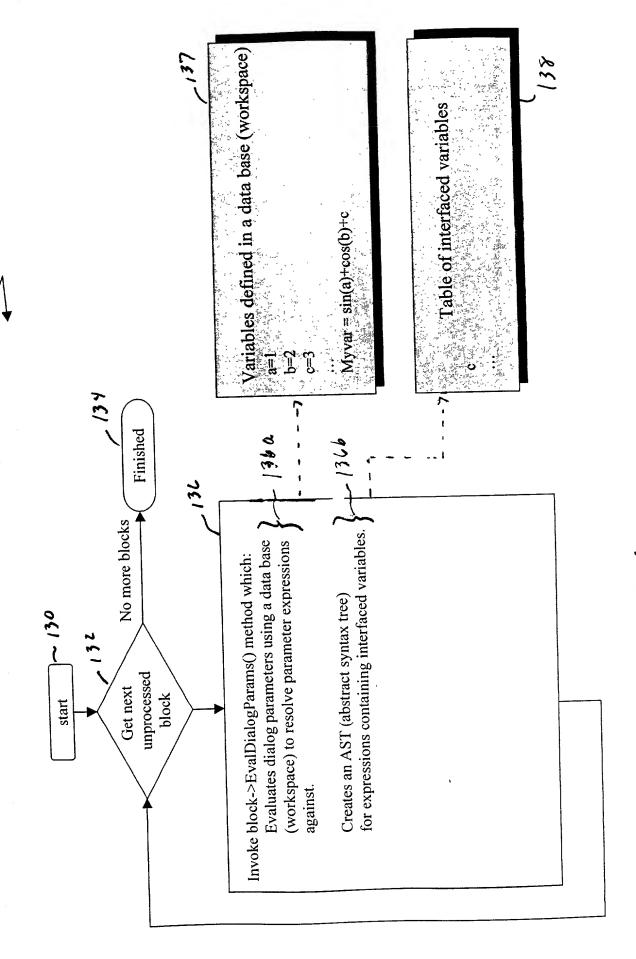


FIG. 7

```
Block {
 EvalDialogParams(void)
 SetupRuntimeParams(void) = 114
                 e- 116
 InternalData:
                         - 118
   AST[nParams];
   NumericDialogParamValues[nParams]; 20
   RuntimeParams[nRunTimeParams]; ~ 122
 GetAstForDialogParam(int paramIndex)
 SetAstForDialogParam(int paramIndex, ast)
 GetNumericParamValues(int dialogParamIndex)
 SetNumericParamValues(int dialogParamIndex, value)
 GetNumRuntimeParams()
 GetRuntimeParam(int runtimeParamIdx) <- 124
 SetRuntimeParam(int runtimeParamIdx, value)
```



F16. 9

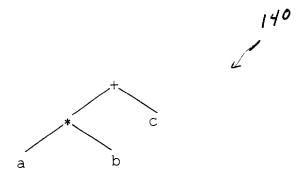


FIG. 10A

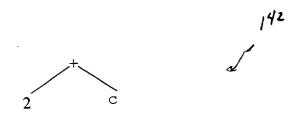
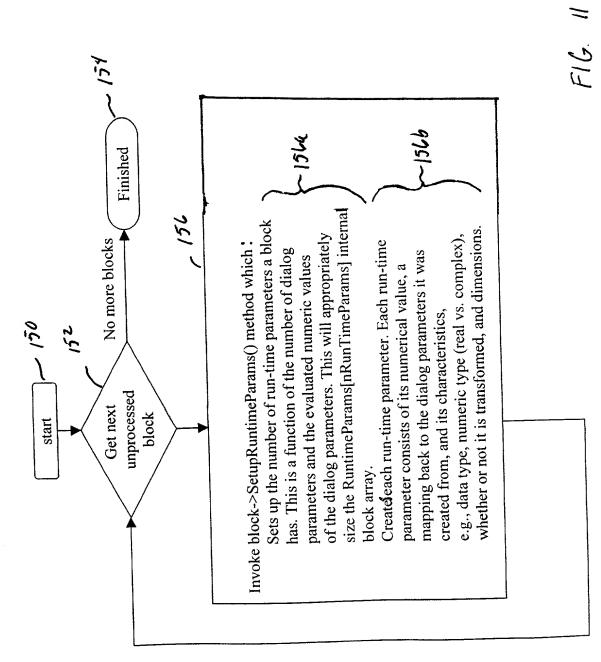
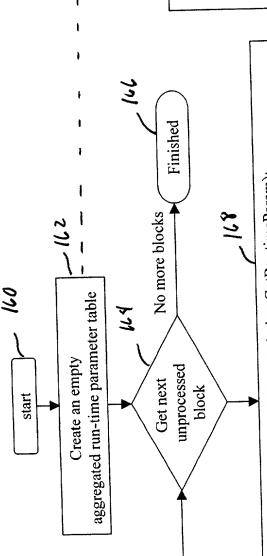


FIG. 10B





Examine each run-time parameter (using GetRuntimeParam);
If run-time parameter does not map back to dialog parameters

containing interfaced variables or if run-time parameter has

been explicitly marked as constant then

cen explicitly marked as constant them

(Compute a checksum of the numeric run-time parameter data

| and using a hash function, see if this value exists in the aggregated run-time parameter table.

168c - If run-time parameter doesn't exist, add it.

1644 \(\) Else, free the blocks numeric run-time parameter data and update the table entry indicating that this run-time parameter is pooled.

I/6 Set the blocks numeric run-time parameter data to be equal

" to the table entry."

Aggregated run-time parameter table

-Uses a hash function on the numerical run-time parameter value to look up to locate entries for non-interfaced run-time parameters.

[1, 2, 3] :: plus back pointer to blocks run-time parameter indices

